

Trouble information extraction based on a bootstrap approach from Twitter

Kohei Kurihara Kazutaka Shimada

Department of Artificial Intelligence
Kyushu Institute of Technology
680-4 Kawazu Iizuka Fukuoka Japan
shimada@pluto.ai.kyutech.ac.jp

Abstract

In this paper, we propose a method for extracting trouble information from Twitter. One useful approach is based on machine learning techniques such as SVMs. However, trouble information is a fraction of a percent of all tweets on Twitter. In general, imbalanced distribution is not suitable for machine learning techniques to generate a classifier. Another approach is to extract trouble information by using handwritten rules. However, constructing high coverage rules by handwork is costly. First, we verify these problems in a preliminary experiment. Then, to solve these problems, we apply a bootstrapping method to our trouble information extraction task. We introduce three characteristics and a scoring method to the bootstrapping. As a result, the iteration process on the bootstrapping increased the number of tweets and patterns for trouble information dramatically.

1 Introduction

The World Wide Web contains a huge number of online documents that are easily accessible. Analysis of the documents has an important role for natural language processing. One of the important information for business companies is trouble information of a product as the risk management. If they can monitor the information about products and the troubles from the Web automatically, they might be able to avoid critical damages by realizing the risk in advance. Therefore trouble information extraction is a significant task in business. There are many studies which handled news articles (Sakai et al., 2006),

review documents (Ivanov and Tutubalina, 2014), financial documents (Leider and Schilder, 2010), daily reports (Kakimoto and Yamamoto, 2008), a failure database on the Web (Awano et al., 2012) and so on, as the target data. However, these information sources are not usually instantaneous and exhaustive. To solve this problem, we focus on Twitter. It is one of the most famous microblogging services and text-based posts of up to 140 characters. The posted sentences are described as “tweets.” We suppose users on Twitter often post tweets with trouble information because they tend to post tweets as lifelog data in real time. Some researchers focused on the characteristic (Aramaki et al., 2011; Sakaki et al., 2010; Shimada et al., 2012).

In this paper, we propose a method to extract trouble information from Twitter. One of the most common approaches is to classify an input into trouble information and non-trouble information by using a machine learning technique. However, most of the tweets do not relate to trouble information. In other word, the ratio of trouble tweets and non-trouble tweets is biased. Such biased data generally generate a unsuitable classifier. Another approach is to extract trouble information by using handwritten rules. However, constructing high coverage rules by handwork is usually a difficult task. In this paper, we investigate these problems through a preliminary experiment. On the basis of the result, we introduce a bootstrapping approach to our trouble information extraction task. Methods based on bootstrapping techniques are one of the effective approaches to extract information (Riloff and Jones, 1999; Etzioni et al., 2004). Riloff et al. (2013) have pro-

posed a method to identify sarcastic tweets by using a bootstrapping algorithm. Ohmori and Mori (2010) have proposed a method based on a bootstrapping approach with words and phrases for searching for failure cases among products. We focus on trouble expressions which indicate the malfunction and failure of products. We apply the trouble expressions as seeds into a bootstrapping approach. By the iteration process, our method obtains more trouble expressions, and then extracts tweets with trouble information.

2 Related work

Trouble identification is one category in sentiment analysis (Pang and Lee, 2008). The classification into trouble or non-trouble is similar to the classification into positive or negative (Pang et al., 2002; Turney, 2002). However, negative opinions are not always equal to trouble information. For example, “I don’t like this product” is a negative opinion, but not trouble information. Therefore, they should be distinguished.

Saeger et al. (2008) have proposed a method to extract object-trouble relations from the Web. They acquired trouble expressions by an unsupervised method, and then classify them by using SVMs. Gupta (2011) has proposed a method to extract problem information using a machine learning technique from Twitter. As the two papers mentioned, the trouble descriptions in the training data were rare, less than 10%. In other words, the ratio of positive and negative instances for this task tends to be biased. Therefore, machine learning approaches are not always suitable for this task.

Solovyev and Ivanov (2014) have proposed a dictionary-based problem phrase extraction from product reviews. It was based on a simple pattern matching with their dictionaries. In (Ivanov and Tutubalina, 2014), they incorporated a clause feature, but-conjunction, with the dictionary-based method. Kakimoto and Yamamoto (2008) have proposed a method based on syntactic pieces for extracting troubles. The basic idea in these studies is similar to our method. However, these approaches did not contain an iteration process like bootstrapping. Although bootstrapping methods often generate noise seeds for the next process and the wrong seeds lead to the

decrease of the precision rate, namely semantic drift, the iteration process is vital to obtain the high recall rate.

Although there are studies based on a bootstrapping approach such as (Leider and Schilder, 2010; Ohmori and Mori, 2010), the targets are not Twitter. Riloff et al. (2013) have handled tweets and used a bootstrapping approach for their task. However, the purpose is to generate a sarcasm recognizer.

3 Trouble information

In this section, we explain the target trouble information in this paper. Here we introduce two words; trouble sentences (TS) and trouble expressions (TE). The TSs are our target in the extraction process. They are tweets with trouble information about a product¹. The TEs are phrases which indicate trouble situation, failure and so on.

TS : Why? My smartphone isn’t powered on....

TE : not powered on

In this paper, a TS needs to contain a product name/information and TE(s). In other words, we do not handle any tweets without a product name/information. In the above instance, “smartphone” is the product name/information. For TEs, we admit figurative phrases, emoticons and Internet slangs. For example, “My phone is dead” and “The home button on iPhone is wroooooong (ToT).”

4 Preliminary experiment

In this section, we describe some problems of a simple machine learning approach and a rule-based approach through an experiment.

4.1 Machine learning based

We constructed a classification model based on SVM (Vapnik, 1995). We used SVM^{light} (Joachims, 1998) for the implementation. Although we utilized some features about emoticons, Internet slang dictionaries and so on, they were not effective. Therefore, we used only the bag-of-words features for SVM.

We prepared 900 tweets for the training data; 450 positive and 450 negative instances. We evaluated

¹The actual tweets in the experiment are written in Japanese.

Recall	Precision	F
0.88	0.98	0.93

Table 1: The experimental result on the leave-one-out cross-validation.

# of EXT	# of COR	Precision
3,742	720	0.19

Table 2: The experimental result for a realistic situation.

the machine learning based method with the leave-one-out cross-validation. Table 1 shows the experimental result. The method produced high recall and precision rates for the cross-validation. However, most of real tweets are non-trouble information. In other words, this situation is not on the real world. Therefore, we also evaluated our method trained by 900 tweets with 30,000 tweets that extracted from Twitter randomly, as an opened test set. This is an real situation, namely unbalance data. We judged the correctness of the outputs of SVM. Table 2 shows the experimental result for the unbalance data. The EXT and COR in the table denote the number of tweets extracted by SVM and the number of tweets extracted correctly, respectively. From the table, the machine learning based method was not suitable for this task because the precision rate on the realistic data set dramatically decreased.

4.2 Rule-based

We also constructed a rule-based method with a simple matching approach. We prepared trouble expressions (TEs) by handwork. Although trouble sentences (TSs) always contain TE(s), all sentences with TEs are not always TSs. Therefore, we also prepared NG phrases for the rule-based method. For example, “can’t charge” is a TE for mobile phones. However, “I can’t charge my phone because I don’t have a charger now” is not a TS because it is not trouble information about a product. To solve this problem, we need to add a NG phrase “because I don’t have a charger.”

We evaluated our rule-based method with 30,000 tweets in Section 4.1. Table 3 shows the experimental result. We obtained high precision rate by using the rule-based method, as compared with the machine learning method (See Table 2.) On the other

# of EXT	# of COR	Precision
474	444	0.94

Table 3: The experimental result of the rule-based method on the same situation with SVM.

hand, the number of tweets extracted correctly was reduced almost by half (720 vs. 444). As a result, the simple rule-based method faced with another problem for this task.

4.3 Discussion

The problem of the machine learning method is caused by the number of tweets and the ratio of positive and negative instances in the training data. The training data with 900 instances was insufficient in terms of the size for machine learning, especially the coverage of non-trouble information. Besides, a classifier in this situation often generates a poor result because the distribution of the training data differs from that of the real data. One intuitive solution is to add new tweets as positive/negative instances. However, collecting tweets with positive/negative by handwork is costly. Moreover, the concrete definition of non-trouble tweets is essentially difficult. Since the realistic situation contains many non-trouble tweets as compared with trouble tweets, the training data should contain many non-trouble tweets. However, combined with the difficulty of the concrete definition of non-trouble tweets, collecting non-trouble tweets with high coverage is also a difficult task. Therefore, machine learning approaches are not appropriate for our task.

The rule-based method obtained the high precision rate. The reason was that we could focus on the trouble expressions in the method as compared with the machine learning method. Although we naturally needed to prepare NG phrases, the effort for the rule-based method was less than that for the machine learning method. Therefore, rule-based methods are essentially appropriate for our task. However, the recall rate was a critical problem for the method. One solution is to increase the number of TEs for the extraction process. On the other hand, constructing TEs with high coverage by handwork is costly. Therefore, we need to extract TEs from tweets automatically.

5 Proposed method

On the basis of the discussion in the previous section, we expand our rule-base method with a bootstrapping approach. The bootstrapping approach leads to the improvement of the coverage of the original rule-based method.

5.1 Outline

For extracting various types of trouble sentences, TSs, it is necessary to acquire new trouble expressions, TEs, automatically. In general, some TEs often appear in one TS. We focus on this characteristic. Figure 1 shows an example. Here, “broken” is a TE, a seed for a bootstrapping approach. Assume that the TE and the phrase “not make a call” often co-occur in tweets. From this observation, our method obtains the phrase “not make a call” as a new TE, and then extract a new TS by the new TE.

The outline of our method is shown in Figure 2. First, we create seed words with strong trouble meanings for a target product by hand. By using the initial seeds, namely TEs, our method extracts TSs from a tweet corpus. For the TS extraction process, we judge the presence of TEs in each sentence. As exceptional treatment, we prepare some non-extraction rules. The non-extraction rules contain hearsay expressions such as “someone told me that” and non-factual expressions such as “feel like.” We do not extract sentences matching with the non-extraction rules as TSs. Next, our method extracts TE candidates from the extracted TSs. For the candidates, we apply a scoring method for computing a confidence measure as new TEs. We acquire only TEs with high confidence values as new TEs. Finally, we add the new TEs to the previous seeds. Our method iterates these processes until it fulfills certain conditions. In this paper, we set two conditions; (1) if the iteration is repeated at 5 times or (2) if the method does not acquire new TEs.

5.2 TE acquisition

TE extraction is based on surface and part-of-speech tags patterns. We focus on the following characteristics for the extraction.

Specific adverbs Adverbs are closely related to trouble information. Murakami and Nasukawa

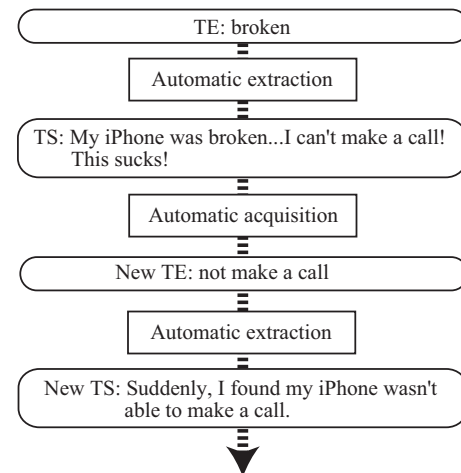


Figure 1: An example of the extraction process.

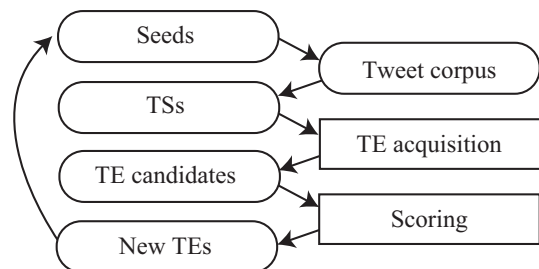


Figure 2: The outline of our method.

(2011) have proposed a method to detect potential problems from documents. They focused on adverbs, such as “suddenly” and “arbitrarily”, to detect the nouns and verbs that described the actual problems. This is a language-independent characteristic. We also extract phrases with the specific adverbs as TEs.

Imperfective forms The target tweets in this paper are written in Japanese. As one Japanese characteristic, TSs often contain the imperfective form of a verb with negation². We extract phrases with this pattern as TEs.

Negative words As we mentioned in Section 2, negative opinions are closely related to trouble information. Tweets with negative expressions have high potentiality for TEs and TSs. On the other hand, as we also mentioned in Section 2, negative opinions are not always equal

²E.g., “動か ない (not work)” and “起動し ない (not start).”

to trouble information. Utilizing general sentiment dictionaries is not always suitable for this task because they contain many negative words not related to trouble information. In this paper, we prepare negative words related to trouble information about a target product, such as “bad”, “wrong” and “failure”, as a negative word set. We extract phrases with the negative words.

5.3 Scoring

A bootstrapping approach uses the previous outputs from the system as the inputs for the system in the next step. If the precision of the outputs is low, it leads to the decrease of the precision of the next outputs. The accuracy deterioration by the change of the meaning of seeds is well-known as “Semantic drift” (Curran et al., 2007). To solve this problem, we need to keep high precision in the iteration process. In other words, we need to reject noise TEs in candidate TEs. Therefore, we need to estimate a confidence measure of each candidate TE.

One of the most successful approaches is the Espresso algorithm (Komachi et al., 2008; Pantel and Pennacchiotti, 2006). The algorithm was based on recursive definition of pattern-instance scoring metrics. It computed the pointwise mutual information between each pattern and instance recursively. The method in this paper does not handle any patterns for the bootstrapping process. Therefore, we cannot incorporate this algorithm into our method directly.

We introduce another scoring method for a confidence measure in the bootstrapping process. First, we compute confidence values of nouns, verbs and adjectives in TSs. Then, we estimate the confidence value of each TE on the basis of the confidence values. The confidence measure is based on the following hypothesis:

- if a word frequently appears in TSs, the TE likelihood of the word is high.
- words appearing near a product name³ contain high TE likelihood.

³It denotes not only concrete product names, such as “iPhone”, but also product categories, such as “smartphone.”

The value of a word w is computed as follows:

$$WS_w = \sum_{i \in I} \frac{1}{dist_i(w)} \quad (1)$$

where i and I are a sentence and sentences including a product name, respectively. $dist_i(w)$ is the distance between a product name and w in i . The confidence measure of a TE_t is the average value of WS_w in the TE.

$$TEscore_t = \frac{1}{N_w} \sum_{w \in TE_t} WS_w \quad (2)$$

where N_w is the number of words in TE_t . If a TE contains frequent words with the high WS_w , it obtains high $TEscore$.

After computation of $TEscore$, we extract phrases in the top $N\%$ as the new seeds for the next iteration. If the phrases in the current top $N\%$ are the same as the phrases in the previous step, the iteration is terminated.

6 Experiment

In this section, we evaluate our method with real tweets, and then discuss the results.

6.1 Result

The target product was cellphones. We collected 100,000 tweets about cellphones as the data set. These tweets contained words that related to cellphones. As initial seeds, we set the following seven words; 壊れる (broken), おかしい (wrong), 異常 (defect), 故障 (defect), フリーズ (freeze) and バグ (bug). We applied the seeds to the data set, and then obtained TEs and TSs by using the proposed bootstrapping method. In this experiment, the total number of iterations was 5. More properly speaking, when the number of iteration was 5, our method did not obtain new TEs. In other words, both of the two conditions in Section 5.1 were fortuitously fulfilled in this iteration.

Figure 3 shows the result of the precision rate and the number of extracted TSs on the iteration. Our method increased the number of TSs in the second step by using new TEs extracted in the first step. Despite the increase of TSs, our method maintained a high precision rate in the second step. After the third step, our method obtained a small increase in

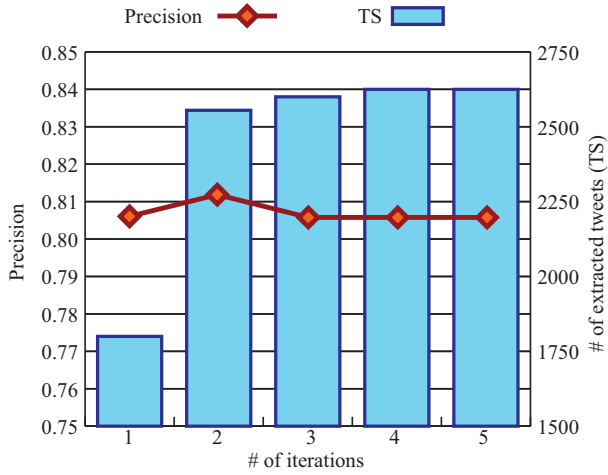


Figure 3: The precision and the number of TSs in each iteration.

terms of the number of TSs and also held the high precision rate. This result denotes that the scoring method in Section 5.3 was effective for the bootstrapping in terms of the noise reduction from candidate TEs. The experimental result shows the effectiveness of our method, as compared with a non-bootstrapping method, because the result in the second step, namely bootstrapping, outperformed that in the first step, non-bootstrapping.

6.2 Analysis and discussion

Our method extracted TSs with a high precision rate through the iteration in the bootstrapping. Table 4 shows instances of TEs extracted by the method. These TEs related to the category “cellphones” and were suitable for the extraction of TSs. Our method correctly extracted some phrases with the opposite meaning, such as “turn on automatically” and “not turn on.” These TEs were distinguished by adverbs; “arbitrarily” and “suddenly.” It is difficult to extract these TEs by using only general sentiment dictionaries. We also obtained domain specific TEs, such as “put the speaker on mute arbitrarily.” Our method extracted various types of TEs by using the bootstrapping method.

Next, we discuss the size of the target data and the accuracy. If the data size is small, our method might not extract sufficient TEs. As a result, it leads to the decrease of the number of TSs. If the data size is large, our method might extract many inap-

勝手に電源がつく (turn on arbitrarily)
電源がつかない (not turn on)
急に電源が落ちる (power off suddenly)
電源が落ちない (not power off)
ボタンが押せない (not push the button)
勝手にスピーカーがミュートになる (put the speaker on mute arbitrarily)

Table 4: Extracted TEs.

# of tweets	# of TSs
10,000	121
50,000	623
100,000	2,623
500,000	9,088

Table 5: The number of extracted TSs on several data sets. The third row is the same as Section 6.1.

propriate TEs. It probably leads to the increase of the number of TSs with non-trouble information and the decrease of the precision. We investigated our method with the different size of data sets. Table 5 shows the result. For smaller data set, namely 10,000 and 50,000 tweets, the number of extracted TSs decreased dramatically. In these data sets, the number of outputs in the first iteration was insufficient. As a result, our method could not obtain TSs and new TEs in the next process. Thus, our method needs an adequate amount of tweets for the TE acquisition process. For a larger data set, 500,000 tweets, the predicted number of TSs was approximately 13,000⁴. The actual number of extracted TSs in the larger data set was 9,088. The result indicates that our method controlled noise TEs appropriately in the bootstrapping process. In addition, our method extracted different types of TEs from the larger data set, such as 勝手にアプリが起動する (an app starts arbitrarily) カードを読み込まない (not recognize a card) and 時計進まない (clock not work). Our method was robust to the increase

⁴It was 13,115 = 2,623 × 5 by simple arithmetic.

of target data and could extract new TEs and TSs efficiently.

Finally, we explain the results of TSs. The following sentences are tweets extracted from our method:

- 充電切れるわケータイ熱くなっちゃって充電できないわ勝手に電源切れるわ最悪です (My cellphone ran out of charge, too hot to charge the battery and power off automatically This sucks!)
- てか携帯画面真っ黒になって電池バック抜いて電源入れようとしても電源つかないんだけど (The display of my cellphone blacked out, I removed the battery, and then I powered on it, but it isn't turned on.)

Although these tweets did not contain direct expressions to trouble information, such as 壊れた (broken), our method correctly extracted them with acquired TEs, such as 勝手に電源が落ちる (power off automatically) and 電源がつかない (not turn on). The following sentence is an incorrect output TS from our method.

- iPhone の充電ケーブル壊れた (; ㄩ ˘) 充電できない (; ㄩ ˘) (The charging cable of my iPhone was broken ... I can't charge the battery.)

This tweet is trouble information about accessories, but not a TS for a cellphone itself. In this experiment, we regarded this kind of outputs as negative results. This is a difficult problem in trouble information extraction. One solution is to add NG rules to the extraction process. However, we cannot solve a problem of the following sentence, which is also a negative result, by addition of NG rules.

- どしたんやろーなんかあったんかな (´ ω ˘) iPhone 壊れたんかな (An accident had happened (to him/her)? ... (his/her) iPhone might be broken⁵.)

This tweet contained a trouble expression, but it is not a TS for a cellphone. It implied that a user was

⁵Note that the question mark and the word “might” in the English translation don't appear explicitly in the original Japanese sentence.

worried about someone. This problem is more difficult because we need deep analysis including semantics to solve it. Handling metaphor and Internet slangs appropriately is also important future work.

In the experiment, we evaluated our method in terms of the precision rate because it is difficult to measure the recall rate. Although we obtained more TSs by using our method, the number of TSs might be insufficient, namely the low recall rate. To improve this problem is the most important issue for our method.

We judged the correctness of the extracted TSs in Figure 3 with one annotator. We prepared a manual for the annotation, such as the definition of trouble information, in advance. However, for more correct and reliable annotation, we need to annotate TSs with several annotators and compute the agreement among them. This is also important future work.

7 Conclusions

In this paper, we proposed a bootstrapping method to extract trouble information from Twitter. As a preliminary experiment, we evaluated a simple machine learning method based on SVM and a simple rule-based method. Although the SVM-based method worked well for the cross-validation about a small data set, the precision rate dramatically decreased for a real and unknown tweet data set. The rule-based method obtained a high precision rate as compared with SVM. However, TSs extracted correctly were reduced almost by half. The main problems of these methods were (1) biased data, (2) coverage about non-trouble information and (3) a limited number of trouble expressions (TEs).

To solve the problems, we applied a bootstrapping approach to the trouble information extraction. By using a small seed set and the bootstrapping approach, our method increased the number of extracted trouble sentences (TSs) by 50% with a high precision rate. We used three characteristics in the TE acquisition; specific adverbs, imperfective forms and negative words. In addition, we introduced a scoring method to avoid the semantic drift problem. The scoring was based on the distance between product information and each word. We verified the effectiveness of our method with different size of data sets. Our method was robust to the increase

of target data and could extract new TEs and TSs efficiently.

In the discussion part of this paper, we explained some problems through the extracted TSs. A simple solution to improve the accuracy is to expand rules for the TE acquisition. In addition, we need to introduce more deep analysis, such as semantic analysis, for the difficult problem described in Section 6.2. We have obtained many tweets with trouble information by our method. Deeper trouble mining from the tweets, such as risk-prone analysis, and visualization of the trouble information are our important future work. Torisawa et al. (2008) have reported a system based on graph drawing as a web search directory. It mapped a topic that a user inputted and the related keywords. This approach is useful to find and understand potential troubles from the extracted TSs. Another useful visualization approach is TreeMap styles (Johnson and Shneiderman, 1991). Carenini et al. (2006) have proposed an interactive multimedia summarization system based on a text summary and a visual summary. Shimada et al. (2010) have reported an interactive multimedia summarization method with the Tree-Map and fisheye-like styles for clustered sentences. The summarization and visualization of the extracted TSs are interesting future work.

References

- Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. 2011. Twitter catches the flu: Detecting influenza epidemics using twitter. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1568–1576.
- Yuki Awano, Qiang Ma, and Masatoshi Yoshikawa. 2012. Cause analysis of new incidents by using failure knowledge database. In *Proceedings of the 23rd International Conference on Database and Expert Systems Applications (DEXA 2012)*, pages 88–102.
- Giuseppe Carenini, Raymond T. Ng, and Adam Pauls. 2006. Interactive multimedia summaries of evaluative text. In *Proceedings of Intelligent User Interfaces (IUI)*, pages 124–131.
- James R. Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (2007)*, pages 172–180.
- Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall (preliminary results). In *Proceedings of the 13th international conference on World Wide Web (WWW2004)*, pages 100–110.
- Narendra K. Gupta. 2011. Extracting descriptions of problems with product and service from twitter data. In *Proceedings of the 3rd Workshop on Social Web Search and Mining (SWSM2011)*.
- Vladimir Ivanov and Elena Tutubalina. 2014. Clause-based approach to extracting problem phrases from user reviews of products. In *Analysis of Images, Social Networks and Texts, AIST 2014*, pages 229–236.
- Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142.
- Brian Johnson and Ben Shneiderman. 1991. Treemaps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd International IEEE Visualization Conference*, pages 284–291.
- Yoshifumi Kakimoto and Kazuhide Yamamoto. 2008. Extracting troubles from daily reports based on syntactic pieces. In *Proceedings of PACLIC 22*, pages 411–417.
- Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in espresso-like bootstrapping algorithms. In *Proceedings of EMNLP 2008*, pages 1011–1020.
- Jochen L. Leider and Frank Schilder. 2010. Hunting for the black swan: Risk mining from text. In *Proceedings of ACL2010 System Demonstration*, pages 54–59.
- Takuma Murakami and Tetsuya Nasukawa. 2011. Detecting potential issues based on typical problem description (in Japanese). In *IEICE, SIG-NLC, 111*, pages 31–35.
- Nobuyuki Ohmori and Tatsunori Mori. 2010. Novel approach for test methods automatic selection in product reliability — improved method for acquiring part-whole relation —. In *Proceedings of International Conference on Machine Learning and Application (ICMLA 2010)*, pages 834–839.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120.
- Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceeding of AAAI 99*, pages 474–479.
- Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of EMNLP 2013*, pages 704–714.
- Stijn De Saeger, Kentaro Torisawa, and Jun’ichi Kazama. 2008. Looking for trouble. In *Proceedings of COLING 08*, pages 185–192.
- HiroYuki Sakai, Shouji Umemura, and Shigeru Masuyama. 2006. Extraction of expressions concerning accident cause contained in articles on traffic accidents (in Japanese). *Journal of Natural Language Processing*, 13(2):99–123.
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World Wide Web (WWW2010)*, pages 851–860.
- Kazutaka Shimada, Masahi Yamaumi, Ryosuke Tadano, Masashi Hadano, and Tsutomu Endo. 2010. Interactive aspect summarization using word-aspect relations for review documents. In *Proceedings of the 5th International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent Systems (SCIS & ISIS 2010)*, pages 183–188.
- Kazutaka Shimada, Shunsuke Inoue, and Tsutomu Endo. 2012. On-site likelihood identification of tweets for tourism information analysis. In *Proceedings of 3rd IIAI International Conference on e-Services and Knowledge Management (IIAI ESKM 2012)*, pages 117–122.
- Valery Solovyev and Vladimir Ivanov. 2014. Dictionary-based problem phrase extraction from user reviews. In *Proceedings of TSD 2014, LNAI 8655*, pages 225–232.
- Kentaro Torisawa, Stijn De Saeger, Yasunori Kakizawa, Jun’ichi Kazama, Masaki Murata, Daisuke Noguchi, and Asuka Sumida. 2008. Torishiki-kai, an autogenerated web search directory. In *Proceedings of the Second International Symposium on Universal Communication (ISUC 2008)*, pages 179–186.
- Peter D. Turney. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL ’02*, pages 417–424.
- Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc.